# Spring Annotations

Spring Annotations are a form of metadata that provides data about a program. Annotations are used to provide supplemental information about a program. It does not have a direct effect on the operation of the code they annotate. It does not change the action of the compiled program.

Basically, there are 6 types of annotation available in the whole spring framework.

1. Spring Core Annotations
2. Spring Web Annotations
3. Spring Boot Annotations
4. Spring Scheduling Annotations
5. Spring Data Annotations
6. Spring Bean Annotations

Spring annotations present in the ***org.springframework.beans.factory.annotation*** and ***org.springframework.context.annotation*** packages are commonly known as Spring Core annotations.

* DI-Related Annotations
  + @Autowired
  + @Qualifier
  + @Primary
  + @Bean
  + @Lazy
  + @Required
  + @Value
  + @Scope
  + @Lookup, etc.
* Context Configuration Annotations
  + @Profile
  + @Import
  + @ImportResource
  + @PropertySource,

@Autowired annotation is applied to the fields, setter methods, and constructors. It injects object dependency implicitly. We use @Autowired to mark the dependency that will be injected by the Spring container.

**1.2:**Field injection

* Java

|  |
| --- |
| **class** Student {      @Autowired      Address address;  } |

**1.3:**Constructor injection

* Java

|  |
| --- |
| **class** Student {      Address address;        @Autowired      Student(Address address) {  **this**.address = address;      }  } |

**1.4:** Setter injection

* Java

|  |
| --- |
| **class** Student {      Address address;        @Autowired  **void** setaddress(Address address) {  **this**.address = address;      }  } |

**Context Configuration Annotations**

@Profile: If you want Spring to use a @Component class or a @Bean method only when a specific profile is active then you can mark it with @Profile.

@Component

@Profile("developer")

public class Employee {}

Spring annotations present in the ***org.springframework.web.bind.annotation p***ackages are commonly known as Spring Web annotations. Some of the annotations that are available in this category are:

* @RequestMapping
* @RequestBody
* @PathVariable
* @RequestParam
* Response Handling Annotations
  + @ResponseBody
  + @ExceptionHandler
  + @ResponseStatus
* @Controller
* @RestController
* @ModelAttribute
* @CrossOrigin

**Example:**@Controller

Spring @Controller annotation is also a specialization of @Component annotation. The @Controller annotation indicates that a particular class serves the role of a controller. Spring Controller annotation is typically used in combination with annotated handler methods based on the @RequestMapping annotation. It can be applied to classes only. It’s used to mark a class as a web request handler. It’s mostly used with Spring MVC applications. This annotation acts as a stereotype for the annotated class, indicating its role. The dispatcher scans such annotated classes for mapped methods and detects @RequestMapping annotations.

**Type 3: Spring Boot Annotations**

Spring annotations present in the ***org.springframework.boot.autoconfigure*** and ***org.springframework.boot.autoconfigure.condition***packages are commonly known as Spring Boot annotations. Some of the annotations that are available in this category are:

* @SpringBootApplication
* @EnableAutoConfiguration
* Auto-Configuration Conditions
  + @ConditionalOnClass, and @ConditionalOnMissingClass
  + @ConditionalOnBean, and @ConditionalOnMissingBean
  + @ConditionalOnProperty
  + @ConditionalOnResource
  + @ConditionalOnWebApplication and @ConditionalOnNotWebApplication
  + @ConditionalExpression
  + @Conditional

 It encapsulates @Configuration, @EnableAutoConfiguration, and @ComponentScan annotations with their default attributes.

**Type 4: Spring Scheduling Annotations**

Spring annotations present in the ***org.springframework.scheduling.annotation*** packages are commonly known as Spring Scheduling annotations. Some of the annotations that are available in this category are:

* @EnableAsync
* @EnableScheduling
* @Async
* @Scheduled
* @Schedules

**Type 5: Spring Data Annotations**

Spring Data provides an abstraction over data storage technologies. Hence the business logic code can be much more independent of the underlying persistence implementation. Some of the annotations that are available in this category are:

* Common Spring Data Annotations
  + @Transactional
  + @NoRepositoryBean
  + @Param
  + @Id
  + @Transient
  + @CreatedBy, @LastModifiedBy, @CreatedDate, @LastModifiedDate
* Spring Data JPA Annotations
  + @Query
  + @Procedure
  + @Lock
  + @Modifying
  + @EnableJpaRepositories
* Spring Data Mongo Annotations
  + @Document
  + @Field
  + @Query
  + @EnableMongoRepositories